

WEST

Freeform Search

Database:

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

cascade and L13

Display:**Documents in Display Format:****Starting with Number****Generate:**☐

Hit List

☒

Hit Count

☐

Side by Side

☐

Image

[Main Menu](#)[Show S Numbers](#)[Edit S Numbers](#)[Preferences](#)[Cases](#)

Search History

DATE: Thursday, April 03, 2003[Printable Copy](#)[Create Case](#)

WEST**Freeform Search****Database:**

US Patents Full-Text Database
 US Pre-Grant Publication Full-Text Database
 JPO Abstracts Database
 EPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Term:

l6 and L7

Display:

10

Documents in Display Format:

KWIC

Starting with Number

1

Generate:☐

Hit List

☒

Hit Count

☐

Side by Side

☐

Image

Search

Clear

Help

Logout

Interrupt

Main Menu

Show S Numbers

Edit S Numbers

Preferences

Cases

Search History**DATE:** Thursday, April 03, 2003[Printable Copy](#)[Create Case](#)**Set Name Query**

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

<u>L8</u>	l6 and L7	4	<u>L8</u>
<u>L7</u>	echo signal and extract\$3	1422	<u>L7</u>
<u>L6</u>	l4 and L5	114	<u>L6</u>
<u>L5</u>	adaptive filter	2812	<u>L5</u>
<u>L4</u>	power same L2	1238	<u>L4</u>
<u>L3</u>	l1 and L2	0	<u>L3</u>
<u>L2</u>	speech signal	10229	<u>L2</u>
<u>L1</u>	5790440.pn.	1	<u>L1</u>

END OF SEARCH HISTORY

WEST

Freeform Search

Database:

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:

19 and cascade

Display:

10

Documents in Display Format:

KWIC

Starting with Number

1

Generate:

☐

Hit List

☒

Hit Count

☐

Side by Side

☐

Image

Search

Clear

Help

Logout

Interrupt

Main Menu

Show S Numbers

Edit S Numbers

Preferences

Cases

Search History

DATE: Thursday, April 03, 2003[Printable Copy](#)[Create Case](#)

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set

DB=USPT; PLUR=YES; OP=ADJ

<u>L15</u>	19 and cascade	14	<u>L15</u>
<u>L14</u>	cascade and l13	10	<u>L14</u>
<u>L13</u>	l11 and L12	117	<u>L13</u>
<u>L12</u>	threshold or reference	2191872	<u>L12</u>
<u>L11</u>	19 and L10	123	<u>L11</u>
<u>L10</u>	calculat\$3 or comput\$3	796494	<u>L10</u>
<u>L9</u>	17 and L8	139	<u>L9</u>
<u>L8</u>	echo (canceler or canceller)	1679	<u>L8</u>
<u>L7</u>	15 and L6	184	<u>L7</u>
<u>L6</u>	coefficient and updat\$3	17867	<u>L6</u>
<u>L5</u>	13 and L4	322	<u>L5</u>
<u>L4</u>	(echo or reflected) signal	15539	<u>L4</u>
<u>L3</u>	11 and L2	1749	<u>L3</u>
<u>L2</u>	power and L1	1749	<u>L2</u>
<u>L1</u>	adaptive filter	2812	<u>L1</u>

END OF SEARCH HISTORY

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set
	<i>DB=USPT; PLUR=YES; OP=ADJ</i>		
<u>L14</u>	cascade and L13	11	<u>L14</u>
<u>L13</u>	l11 and L12	112	<u>L13</u>
<u>L12</u>	calculat\$3 or comput\$3	796494	<u>L12</u>
<u>L11</u>	l9 and L10	127	<u>L11</u>
<u>L10</u>	echo (canceler or canceller)	1679	<u>L10</u>
<u>L9</u>	l7 and L8	143	<u>L9</u>
<u>L8</u>	subtract\$3	146725	<u>L8</u>
<u>L7</u>	l5 and L6	157	<u>L7</u>
<u>L6</u>	coefficient and updat\$3	17867	<u>L6</u>
<u>L5</u>	l3 and l4	264	<u>L5</u>
<u>L4</u>	power	937812	<u>L4</u>
<u>L3</u>	l1 and l2	409	<u>L3</u>
<u>L2</u>	echo signal	7141	<u>L2</u>
<u>L1</u>	adaptive adj filter	2812	<u>L1</u>

END OF SEARCH HISTORY

WEST

Generate Collection

L15: Entry 14 of 14

File: USPT

Sep 13, 1988

DOCUMENT-IDENTIFIER: US 4771396 A

TITLE: Digital filters

Abstract Text (1):

A transversal finite impulse response, adaptive digital filter comprises three stages (10, 11, 12). The stage (10) is a memory stage which stores input samples x and coefficients. The second stage (11) is an arithmetic stage which forms convolution products and accumulates the product to provide an output. The third stage (12) forms update values for updating the coefficients according to a predetermined algorithm. The stages are coupled by buses (14, 15) and data transfer between the stages via the buses is controlled by a control unit (18). The filter operates so that convolution products are formed during a first part of each sample period and update values are formed during a second part of each sample period for use in the subsequent sample period.

Brief Summary Text (1):

This invention relates to digital filters. Digital filters are devices which receive digital input samples, operate on the samples by multiplying them by filter coefficients to form convolution products and accumulate the products to produce an output. The filter coefficients are selected to produce a desired impulse response.

Brief Summary Text (2):

In applications where the filter is used to model a path, eg an audio path, it can be made adaptive. In this case the output signal of the filter is compared with a desired signal to produce an error signal and the error signal is used to adjust the filter coefficients so that the error signal tends to zero.

Brief Summary Text (4):

Another problem which occurs in conventional adaptive digital filters is what is known as a limit cycle. In an audio application the filter receives audio samples at intervals referred to as audio cycles. During each audio cycle there are a plurality of process cycles during each of which a convolution product is formed. In conventional filters all the process cycles which make up an audio cycle are used to form convolution products, the error signal derived from that audio cycle not being formed until the next audio cycle so that the up-dated filter coefficients are not available until the subsequent audio cycle ie two cycle delay. This delay is known as limit cycling and affects the speed of operation of the device.

Brief Summary Text (5):

According to one aspect of the present invention there is provided an adaptive digital filter comprising a memory stage for storing input samples and filter coefficients, an arithmetic stage for forming convolution products of said input samples and said coefficients and accumulating said products to provide an output signal, a third stage arranged to form an update signal for adapting said filter coefficients according to a predetermined algorithm, bus means coupling said stages to permit data transfer therebetween, and control means for providing timing signals to control operation of said stages and transfer of data between said stages.

Brief Summary Text (7):

According to another aspect of the present invention there is provided an adaptive digital filter of the type which receives input samples x, forms during each sample period, a plurality of convolution products which are accumulated to produce an output signal, and adjusts the filter coefficients according to a predetermined algorithm wherein said convolution products are formed during a first part of each sample period and said filter coefficient adjustment is performed during a second

part of each sample period which is subsequent to said first part.

Brief Summary Text (9):

According to a further aspect of the present invention there is provided an adaptive digital filter of the type which receives input samples x , forms during each sample period a plurality of convolution products which are accumulated to produce an output signal, and adjusts the filter coefficients according to a predetermined algorithm wherein said filter includes means for forming during each sample period values of $x_{sup.2}$ which are used in forming an update value for said filter coefficients according to said algorithm.

Brief Summary Text (10):

According to a still further aspect of the present invention there is provided a method of operating an adaptive digital filter of the type which receives input samples x , forming during each sample period a plurality of convolution products which are accumulated to produce an output signal, and adjusting the filter coefficients according to a predetermined algorithm, the method comprising the steps of forming convolution products during a first part of each sample period, and calculating adjustment values of the filter coefficients during a second part of each sample period which is subsequent to said first part.

Detailed Description Text (1):

Referring to FIG. 1 an audio band adaptive, finite impulse response digital filter which can be implemented in LSI can be considered as comprising three functional blocks 10, 11 and 12 which are interconnected by two 16 bit data buses 14, 15. The block 10 is a memory block which can store 12-bit samples of an audio band input signal x which is to be filtered together with a plurality of filter coefficients h , stored to 16 bit precision, which are to be multiplied with the input samples to form the convolution products. Two sets of filter coefficients are stored, one being known as the adapt coefficients which can adapted during each audio cycle and the other being known as a store version which is not adapted. The memory block 10 comprises three circulating shift registers, one for storing the twelve-bit x samples, a second for storing the adapt coefficients and the third for storing the store coefficients.

Detailed Description Text (2):

The block 11 is a multiplication and accumulation block in which the products of the samples x and the filter coefficients are formed and then accumulated to form the required or replica signal which is output to the block 12. The convolution products for the adapt coefficients are formed and accumulated separately from those for the store coefficients. The two sets of accumulated values are used to produce the replica signal. In addition in the block 11 a quantity known as the sum of the squares of x (SSX) is formed and the significance of this will become apparent later.

Detailed Description Text (3):

The block 12 is a convergence device which inter alia calculates update values for updating the filter adapt coefficients stored in the block 10 according to a predetermined algorithm.

Detailed Description Text (5):

A feature of the present filter is the way in which the convolution products, the replica signal and the update values are evaluated. A conventional filter operates in cycles which in a telephone application will be audio cycles. A typical audio cycle will comprise 256 process cycles. In such conventional filters all 256 process cycles are used in the formation of convolution products and these products are not used in the formation of the error signal until the next audio cycle such that the calculated error signal and hence the updated filter coefficients are not available until the following audio cycle which is thus two cycles later. This leads to a delay which is known as limit cycling. The introduction of a limit cycle has a number of consequences. A minor consequence is the need for an additional storage element which is required to hold certain values but more importantly is the reduction in the rate of adaptation of the filter.

Detailed Description Text (6):

In the present filter each audio cycle is considered as comprising two parts or periods one of which is called the burst period and the other of which is called the pause period (see FIG. 7). During the burst period the convolution products are formed and during the pause period the replica signal is computed together with the error and update signals. Thus the updated coefficient values are available for the next audio cycle. It will be noted that the number of process cycles available for convolution products is reduced and hence the number of coefficients which can be used is also reduced compared with a conventional filter. However, this is considered to be a minor point which compared with the speeding up of the rate of adaptation which is obtained by dividing each audio cycle as described above. In fact, a much more accurate filter is realised.

Detailed Description Text (8):

The filter described with reference to FIG. 1 effectively comprises two filters, one of which makes use of the adapt coefficients and the other which makes use of the store coefficients. In operation the two sets of coefficients and the samples x are multiplexed onto the buses 14, 15 and into the block 11 so that the convolution products of two separate impulse responses with the samples x can be formed and accumulated. The coefficients in one store can be transferred to the other and vice versa.

Detailed Description Text (10):

The type of structure described with reference to FIG. 1 has particular application in a loudspeaking telephone. Such an application is shown schematically in FIG. 2. In this arrangement, digital samples x of an audio signal x (the loudspeaker signal) are fed through filters 25, 26 which have adapt and store coefficients respectively. The filters are arranged to model a path h through which the audio signal are transmitted and to produce a signal for cancelling the "echo" E from the microphone. The filter 26 is a normal transversal filter which is located in the transmission path (or foreground) and the coefficients for this filter are obtained by transfer from the adapt (or background) filter 25.

Detailed Description Text (11):

Coefficient transfer from the background filter 25 to the foreground filter 26 takes place under the control of a microprocessor 28 when a comparison of the error output from each filter indicates that the background filter has a better performance than the foreground. This arrangement can track a changing impulse response rapidly by means of the adaptive filter but allows performance measurement before the previous estimate of impulse response is deleted.

Detailed Description Text (12):

The following theoretical analysis will enable those skilled in the art to appreciate the manner in which the present filter functions. The convolution products are formed and accumulated in accordance with the well know relationship $y = \sum h_n x_{n-k}$ where y is the output signal and the coefficients h are selected to provide the required filter impulse response.

Detailed Description Text (13):

The update values are based on a distributed normalisation architecture which uses a least mean squares algorithm. In this analysis the following symbols are used:

Detailed Description Text (15):

Δh update component

Detailed Description Text (16):

ΔH vector of update component

Detailed Description Text (19):

h coefficient of impulse response which is to be modelled by the filter

Detailed Description Text (21):

h^* estimate of impulse response coefficient--ie the coefficient actually used in the filter

Detailed Description Text (34):

.mu. scaler for update term .DELTA.h

Detailed Description Text (35):

w number of coefficients in impulse response

Detailed Description Text (37):

C reciprocal of the number of cascade stages

Detailed Description Text (43):

In FIG. 3 \hat{x} is an analogue input by definition of infinite precision. Assuming that the reference channel impulse response is bounded by the product of sample period T and number of coefficients W, the error, in vector notation, at sample \hat{j} using Equation (1) is: \hat{X}_j where X_j is the vector of x at time instant j (containing terms $(x_{\text{sub}.j}, x_{\text{sub}.j-1} \dots x_{\text{sub}.j-W-1})$).

Detailed Description Text (47):

Differentiating Equation (2) w.r.t. H^* and substituting into Equation (3). \hat{X}_j The second term in equation (4) represents the update value for updating the filter coefficients at the end of an audio cycle.

Detailed Description Text (48):

The update vector requires two multiplications, applying Equation (1) and substituting for the misadjustment vector: \hat{X}_j Then \hat{X}_j Form the square of the Euclidean vector length: \hat{X}_j Take expectations and assume that the noise contributions are independent random deviates with zero mean and variance σ^2 .

Detailed Description Text (50):

All noise components with the exception of \hat{d} found from the expansion of Equation (2) are included in \hat{n} . The term $E(\hat{n} \cdot \hat{n})$ will be retained since for speech-like signals an estimate the power of the input signal is difficult.

Detailed Description Text (55):

The LSI uses fixed point arithmetic, except for $f(2\mu)$. \hat{g} is specified by the dynamic range of the A-D conversion and expected reference path gain. Using 12 bit conversion and a lossless path $g=2^{11}$. To determine the maximum step-size, it is seen from Equation (13) that the minimum misadjustment occurs when the noise contributions from bus truncation tend to zero. \hat{X}_j Estimating speech signal power requires a pessimistic value to ensure stability but the instantaneous value of $\hat{n} \cdot \hat{n}$ (SSX) may be simply calculated. The maximum $\hat{n} \cdot \hat{n}$ value is dependent on the minimum peak to mean ratio of \hat{x} and the number of coefficients. To handle full-scale sine waves when Equation (14) is unity then: \hat{X}_j For a single filter module with $W \approx 2^{18}$, $g=2^{11}$ and $k=1$, Equation (15) equals 2^{18} which must be distributed between the normalisation of each multiplication.

Detailed Description Text (57):

As has been described above with reference to FIG. 1 the filter unit is partitioned into three functional blocks connected to a thirty-two bit data bus, an eight bit control bus plus serial I/O and clock lines. The memory 10 consists of three circulating shift-registers, one contains the twelve bit X vector, the other two hold H^* adapt and store to sixteen bits. This finite precision is limited by the external bus size (which may be extended) and determines the quantisation noise of $\sigma_{\text{sub}.h}^2$. The choice of shift-register storage reduces the problem of data read/write to one of generating suitable clock strobes. Transfer of data between H stores is signalled on the control bus as update or download to direct data between shift-registers within a period T. To load an arbitrary impulse response into the stored filter requires an external circuit which writes to the adapt filter, and downloads the response into the stored filter.

Detailed Description Text (59):

A number of functions are performed in the convergence device dependent on the control bus word, which is read every period T. The control bus provides the following facilities: cascade depth, speed of adaptation \hat{q} , update, download,

zero, freeze the `adapt` filter and output the H^* overflow status. $f(2\mu.)$ is calculated from SSX (Equation (16)) and the error is the desired signal (y) minus the replica. These values are input to a 12 by 12 bit ripple-through multiplier and the output is then scaled by `k` with a barrel shifter and latched back into one multiplier input port during the `pause`. It should be noted that the internal arithmetic is two's complement which requires rounding to remove d.c. offset after truncation.

Detailed Description Text (60):

The bus data skew enables the coefficient update term

$\text{.DELTA.h} = f(2\mu.) \cdot \text{multidot.e} \cdot \text{multidot.x}$ to be formed during the `burst` in the multiplier, scaled in a barrel 62 shifter by $c.q$ and added to the correct value of h^* before its use in the convolution. Division of .DELTA.h by `Q shifting` is implemented as:

Detailed Description Text (63):

Referring now to FIG. 4 of the drawings the block 10 is shown as comprising an X store 30 which can receive input signals x via a SIPO 31 and a multiplexer 32. The block also includes a H store 33 for storing the filter coefficients. The x values and the h^* values are clocked out from the respective stores onto the buses 14 and 15 and fed to the block 11. The block 11 includes a multiplier 35 which receives the x values via a delay element 36 and the H values via the bus 14. The multiplier 35 forms the convolution products which are summed in an accumulator 136. As will be seen later, the accumulator 136 actually contains two summing circuits, for the foreground and background filters respectively though the multiplier 35(41) is multiplexed. The output of the summing circuit 36 is applied to a circuit 37 which shifts the signals according to the values g and k . The values are fed into respective parallel input serial output circuits 38, the outputs of which constitute the required replica signals r . This signal is fed to the block 12. The block 11 also includes a latch 40 which feeds x values to a circuit 41 which forms values of $x_{\text{sup.2}}$. These are accumulated in an accumulator 43, scaled in a circuit 44, and fed into a PISO 45. The output of the PISO 45 is the required signal SSX which is fed to the block 12 for the convergence calculation. The block 12 receives the desired signal y which is fed via an SIPO 101 to a summation circuit 51 where it is combined with the negative of the replica signal r to form the error signal on line 53. The error signal e is applied to a multiplier 55 which combines the signal e with the signal $f(2\mu.)$ derived from the SSX signal by means of an encoder 54 (as described in more detail below) to form the product $f(2\mu.)e$ on line 56. This is scaled by a barrel shifter 57 and applied to a latch 58. The output from the latch 58 is multiplied in a multiplier 61 by the x value on bus 15 to form a value $f(2\mu.)e \cdot \text{multidot.x} \cdot \text{multidot.k} / t_{\text{sub.1}}$. The output from the multiplier 61 is applied to a barrel shifter 62 to provide the update signal .DELTA.h on line 63 which is summed in an adder 64 to provide the updated value of the coefficient which is then fed into the store 33.

Detailed Description Text (64):

The operation of the filter will be described now in more detail by considering the sequence of data flow for an input signal $x_{\text{sub.n}}$. The signal $x_{\text{sub.n}}$ will be followed from its acquisition by the filter through one audio cycle in a convolution where it will contribute to an error signal and finally be used to update the correct impulse response coefficient in the same cycle.

Detailed Description Text (65):

Consider an input signal $x_{\text{sub.n}}$ which has been obtained from the external environment together with a desired (or echo) signal $y_{\text{sub.n}}$ at time interval n . These signals are applied serially to the filter the x input being fed to the memory block 10 and the y input to the convergence block 12. The x input is clocked into the SIPO 31 prior to process cycle 253 (see FIG. 6--line 1) where it overwrites the oldest value of x which is stored in the rotating shift register, that is $x_{\text{sub.n--257}}$. Simultaneously the new value of x appears on the bus 15 during the high part of strobe .OMEGA.23 in process cycle 253 (see waveform 7 in FIG. 6). Execution of convolution with $x_{\text{sub.n}}$ now takes place in the $(n+1)$ th audio cycle so that one audio cycle delay is introduced.

Detailed Description Text (66):

$x_{sub.n}$ now forms part of the first element in the two convolutions of audio cycle $n+1$ where it is multiplied by $h_{sub.s.sup.0}$ and $h_{sub.a.sup.0}$ respectively which occur on the bus 14 during process cycle zero. This multiplication is carried out in block 35. To begin the formation of updating SSX, the three process cycle delay of $h_{sub.s}$ from the x value enables $x_{sub.n}$ to be extracted early, ie bypassing the built in delay on the convolution (multiplier-accumulator 35,136) device and applied to both sides of a multiplier 41 (in practice, a single multiplier may be used to carry out the functions of multipliers 41 and 35) to form $x_{sub.n.sup.2}$ at the output which is then added to the SSX accumulator 43 to form part of equation 17.

Detailed Description Text (70):

At some time prior to process cycle 251 a new desired echo signal $y_{sub.n+1}$ was entered into a SIPO 101 which shifted the signal $y_{sub.n}$ into a second SIPO 102. Thus the one audio cycle delay in sampling the external signals is accounted for. The number format of the $y_{sub.n}$ signal in SIPO 102 is changed to two's complement by a converter 103 and then applied to the B input of an adder 104. The A input of the adder 104 receives the replica signal which has been overflow checked and complemented (140) together with setting the carry in of the adder to one implementing a sign change. The output of the adder 104 is the error signal e which is fed into two directions. In one direction the number is reformatted in a block 105 and output to a PISO 106 being latched on a pulse PL1 in process cycle 252 to 253. This allows the error signal to be output to the external environment. The other path for the error signal is via the A input of multiplexer 108 which is clocked on pulse MPX 1+3 (see waveform 14 in FIG. 6) onto the B input of a multiplier 110. It should be noted that the important strobe pulse is $.\alpha_{sub.3}$ in process cycle 252 which is applied to the B input latch of the multiplier 1109 and the P output latch of the multiplier.

Detailed Description Text (73):

At this time the A input to the multiplier 110 receives the value $x_{sub.n}$ from the bus 15 via the A input of the multiplexer 125 which contains the value $x_{sub.n}$ during the pulse $.\alpha_{sub.3}$ in the process cycle 254. Thus, the input to the multiplier 110 has both terms to form the first update value which is

Detailed Description Text (74):

This value is latched out from the P latch of the multiplier 110 on the pulse $.\alpha_{sub.3}$ in process cycle 255. It will be seen that the multiplier 110 in fact performs both the multiplication functions 55, 61 of FIG. 4. The final stage of normalisation is applied at the multiplier output by selecting 17 bits and this is equivalent to dividing by $t_{sub.2}$ and truncating. This introduces a noise component $n_{sub.h}$ (see Equation 6). The data value is checked for overflow and presented to the input of the barrel shifter 129. The barrel shifter has decoded the setting of Q shift and the number of cascade stages $1_{sub.c}$ from the control bus. The result is latched by latch 130 on the pulse $.\alpha_{sub.1}$ in process cycle 0 and presented to the A input of a 16 bit adder 64. The least significant bit is applied to the carry-in pin to implement rounding. At this time the bus 15 contains $h_{sub.a.sup.0}$ from the adapt filter which is clocked into a latch 138 on the pulse $.\alpha_{sub.1}$ in process cycle 0 and presented to the B input of the adder 135. The output from the adder is thus the updated value of $h_{sub.a.sup.0}$ and is checked for overflow before being placed on the bus 14 via a tristate buffer 140 during strobe $.\Omega_{sub.23}$ in process cycle 0.

Detailed Description Text (75):

On pulse $.\alpha_{sub.3}$ in process cycle 0 the updated $h_{sub.a.sup.0}$ is latched into the multiplier 35 of the convolution device 11 to multiplied by $x_{sub.n+1}$ and thereby begin the next convolution process. It is also latched into the adapt register of the memory device 10 so that it can be output as $h_{sub.1.sup.0}$ old on the bus 15 in the following audio cycle.

Detailed Description Text (76):

It will thus be seen that by stealing a minimum of three process cycles at the end of each audio cycle the algorithm can be implemented without introducing a limit cycle. In the first of these cycles the factor $f(2_{sub.\mu})$ and the error e are calculated; in the second the product $f(2_{sub.\mu}).\text{multidot}.e$ is formed. With some parallel processing the update component for the first value in the impulse response

Detailed Description Text (81):

A number of filters can be cascaded to increase the effective number of coefficients. FIG. 8 shows the cascading of four filters 201, 202, 203, 204. In order to cascade filters it is necessary to pass the correct `x` data to each unit and accumulate externally the partial SSX and replica terms (adders 205, 206 respectively) returning the total values to each filter unit as shown in FIG. 8. Note that the r and SSX lines need to be broken as shown at A and B in FIG. 4.

Detailed Description Text (84):

The hardware solution is slightly different for SSX. Since the signal x will, in general, be continuous, it can be assumed that the total energy is a quad filter will be 4 times that in a single section. Hence, the final value of SSX can be divided by 4, ie selecting the upper 12 bits from the 14 bit word. The scale factor .sup.1 /c (the level of cascading) is then corrected on the convergence device. This approach allows for maximum utilization of the 12-bit compensation range (if the above assumption is valid). The delay introduced is one process-cycle per level of cascade (see waveform DSTBX); a half process-cycle for serial addition, the other half to divide by two.

Other Reference Publication (1):

B. Widrow, "Adaptive Filters" in Aspects of Network and System Theory", edited by R. E. Kalman and N. De Claris pub. Holt, Rinehart and Wilson, ISBN 03-0772200-6; Relevant as General background on the subject of Adaptive Filters.

Other Reference Publication (2):

K. Ochia, T. Araseki and T. Ogihara, "Echo Cancellor with Two Echo Path Models," IEEE Transactions on Communications, vol. 25, No. 6, pp. 589-595, Jun. 1977, describes a two echo path modem.

Other Reference Publication (3):

B. V. Rao and T. Murali, "A New Design for Digital Adaptive Filters", Int. J. Electronics, vol. 55, No. 3, pp. 473-477, 1983; discusses the problems of limit cycling.

Other Reference Publication (4):

K. Takahashi, T. Sakamoto and I. Tokizawa, "A Cascadable Echo Cancellor" Int. Conf. Commun., 1982 Jun. 13-17, Philadelphia, (IEEE).

Other Reference Publication (6):

IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-31, No. 3, Jun. 1983, pp. 541-549, New York, U.S.: C. F. N. Cowan et al.: "A Digital Adaptive Filter Using a Memory-Accumulator Architecture: Theory and Realization".

Other Reference Publication (7):

ICASSP 84 Proceedings, vol. 1, Mar. 19-21, 1984, San Diego, Calif., U.S., pp. 3.4.1-3.4.4, IEEE; C. R. South et al.; "Extension Facilities and Performance of an LSI Adaptive Filter".

CLAIMS:

1. An adaptive digital filter comprising means (10) for receiving and storing input samples (x) during recurring sample periods, means (11) for forming, during each sample period from said input samples and filter coefficients, a plurality of convolution products and for accumulating said plurality of products an output signal (r), and means (12) for adjusting said filter coefficients according to a predetermined algorithm, the forming means forming the said convolution products during a first part of each sample period and the adjusting means adjusting said filter coefficients, in response to said convolution products formed during said sample period first part, at least in part during a second part of each sample period which is subsequent to said sample period first part.

2. A digital filter as claimed in claim 1 wherein:

said receiving and storing means includes a memory stage (10) for storing input samples (x) and the filter coefficients (h),

said forming and accumulating means includes an arithmetic stage (11) for forming convolution products of the input samples and the said coefficients and accumulating the products to provide an output signal (r), and

said adjusting means includes means (12) for forming an update signal (h) adapting the filter coefficients according to a predetermined algorithm.

3. A digital filter as claimed in claim 2 wherein said memory stage (10) comprises a first section (30) for storing values of input samples (x), a second section (part of 33) for storing values of filter coefficients (ha) which can be adapted, and a third section (part of 33) for storing values of filter coefficient (hs) which are not adaptable.

5. A digital filter as claimed in claim 3 or 4 wherein the arithmetic stage includes means (35) for forming a first set of convolution products between values of the input signal (x) and filter coefficients (ha) from the second memory section, means (35) for forming a second set of convolution products between values of the input signal (x) and filter coefficients (hs) from the third memory section, and means (136) for accumulating said first set of convolution products to form a first output signal and for accumulating said second set of convolution products to form a second output signal.

7. A digital filter as claimed in claim 6 wherein the input samples (x) and coefficient values (h) are multiplexed onto the bus means (14,15) for transfer from the memory stage (10) to the arithmetic stage (11) under the control of clock signals from said control means (18,20).

8. A filter as claimed in any one of claims 2 to 3 wherein:

said adjusting means updates values for the filter coefficients by adding said preset coefficients to a correction factor equal to the product of (a) a scaling factor, (b) an error factor representing the difference between the filter output and a desired signal, and (c) a respective one of the last W input samples, where W is the number of coefficients; and

said forming and accumulating means performs a multiplication operation on the input samples to obtain said convolution products and adds said convolution products to said coefficients during the first part of the sample period, and said adjusting means multiplies the scaling factor by the error factor during the second part of the preceding sample period.

9. A digital filter as claimed in claim 8 wherein said arithmetic stage includes means (41,43) arranged during each sample period to form from sample values x values of $x_{sup.2}$, accumulated over a predetermined number of samples, which are fed to the third stage (12) for use in forming said update values, and wherein the said predetermined algorithm for updating the filter coefficients H can be expressed as

$$H_{sub.new} = H_{sub.old} + f(2.\mu.) \cdot e \cdot x$$

where e is an error signal equal to the difference between the filter output (r) and a reference signal (y), x is the input sample and $f(2.\mu.)$ is a factor derived by decoding in said third stage (12) the said accumulated values of $x_{sup.2}$.

10. A filter according to claim 9 in which,

(a) in the second part of the (n-1)th sample period, the scaling factor $f(2.\mu.)$ is derived from the accumulated value of $x_{sup.2}$; a new sample (x) is received, and the square of this new value is added to the accumulated value of $x_{sup.2}$, and the error between the filter output and the derived value is obtained and its product with $f(2.\mu.)$ obtained,

(b) in the first part of the nth sample period, a plurality W of process cycles

occur in each of which a respective one of the stored sample values x , $x_{\text{sub}.n-1}$. . . $x_{\text{sub}.n-w-1}$ is multiplied by the said product and added to the respective one of the filter coefficients; and the updated value of the coefficient is multiplied by the corresponding one of the stored sample values and accumulated, whereby at the end of the first part of the n th sample period an accumulated filter output is obtained using the error obtained from the immediately preceding coefficient update and

(c) in the second part of the n th sample period the sample value $x_{\text{sub}.n-w}$ is squared and subtracted from the accumulated $x_{\text{sup}.2}$ value.

13. A digital filter which comprises a plurality of filters according to claim 1 arranged in cascade.

14. An adaptive digital filter comprising a memory stage for storing input samples and filter coefficients, an arithmetic stage connected to said memory stage for forming convolution products of said input samples and filter coefficients and for accumulating said products to provide an output signal during a predetermined sampling period, a third stage arranged to form an update signal for adapting said filter coefficients according to a predetermined algorithm substantially during said same predetermined sampling period, bus means coupling said stages to permit data transfer therebetween, and control means for providing timing signals to control operation of said stages and transfer of data between said stages.

15. An adaptive digital filter comprising:

means for sampling an input signal during recurring sampling periods each including a first portion and a second portion subsequent to said first portion;

means connected to said sampling means for forming convolution products during said sampling period first portions in response to said sampled input signal and preset filter coefficients, and for generating a filter output signal in response to said convolution products; and

means connected to said forming means for adjusting said preset filter coefficients during said sampling period second portions,

said adjusting means adjusting, substantially during the second portion of a given sampling period, said coefficients in response to convolution products formed from input signal samples obtained during the first portion of said same given sampling period, said adjusting means making said adjusted coefficients available at least by the time said first portion of the sampling period immediately succeeding said given sampling period occurs.

16. In an adaptive digital filter of the type which receives and samples a digital input signal, multiplies said input signal samples by filter coefficient values to form convolution products, and accumulates the convolution products to produce a filtered output, an improvement comprising:

(a) forming convolution products during an initial, burst portion of a first sampling period in response to a digital input signal sample associated with said first sampling period and in response to said filter coefficient values;

(b) adjusting said filter coefficient values in response to said convolution products formed by said forming step during said first sampling period burst portion substantially during a pause portion of said first sampling period subsequent to said burst portion; and

(c) forming convolution products during an initial, burst portion of a second sampling period immediately succeeding said first sampling period in response to a digital input signal sample associated with said second sampling period and also in response to said filter coefficient values adjusted by said adjusting step (b).

17. In an adaptive digital filter of the type which receives and samples a digital input signal, multiplies said input signal samples by preset filter coefficient

values to form convolution products, and accumulates the convolution products to produce a filtered output, an improvement comprising:

means for forming convolution products during an initial, burst portion of a first sampling period in response to a digital input signal sample associated with said first sampling period and also in response to said filter coefficient values; and

means for adjusting said filter coefficient values, in response to said convolution products formed by said forming means, substantially during a pause portion of said first sample period subsequent to said burst portion,

wherein said forming means also forms convolution products during an initial, burst portion of a second sampling period immediately following said first sampling period in response to a digital input signal sample associated with said second sampling period and also in response to said filter coefficients adjusted by said adjusting means during said first sampling period pause portion.